Homework 4 - Strings and Lists

CS 1301 - Intro to Computing - Spring 2022

Important

- Due Date: **Tuesday, February 15th, 11:59 PM**.
- This is an individual assignment. High-level collaboration is encouraged, **but your submission must be uniquely yours.**
- Resources:
  - TA Helpdesk
  - Email TA's or use class Piazza
  - How to Think Like a Computer Scientist
  - CS 1301 YouTube Channel
- Comment out or delete all function calls. Only import statements, global variables, and comments are okay to be outside of your functions.
- **Read the entire document before starting this assignment.**

The goal of this homework is for you to enhance your understanding of strings, and lists. The homework will consist of 5 functions for you to implement. You have been given `HW04.py` skeleton file to fill out. However, below you will find more detailed information to complete your assignment. Read it thoroughly before you begin.

Hidden Test Cases: In an effort to encourage debugging and writing robust code, we will be including hidden test cases on Gradescope for some functions. You will not be able to see the input or output to these cases. Below is an example output from a failed hidden test case:

```
Test failed: False is not true
```

Written by Alexa O'Reilly (aoreilly@gatech.edu) & Lasya Sreenivasan(lsreenivasan6@gatech.edu)

Message Decoder

Function Name: messageDecoder()
Parameters: message (`str`)
Returns: decodedMessage (`str`)
Description: While on the run from Voldemort, Harry, Ron, and Hermione have been in contact with the Order to update them on their mission and receive updates from them as well. In the case that their messages get intercepted, they have been using encoded messages to communicate with the Order. In order to decode the message, they have to remove any character that is not a letter or a space and reverse the order of the message. Create a function that takes in an encoded message and returns the decoded version of the message. If there are no letters or spaces in the message, return the string `'No message'`.

Hint: .isalpha() could be helpful for this question.

```
>>> messageDecoder('eFa@s9 E5&%r4a $45e\w')
'we arE saFe'
```

```
>>> messageDecoder('Ol2$l@eh')
'hellO'
```

Hermione's Club Members

Function Name: clubMembers()
Parameters: interested (`list`), recruits (`list`)
Returns: memberList (`list`)
Description: Hermione is recruiting people to join S.P.E.W and has one large list of interested people and one smaller list of people she recruited that week. In order to be an official Hogwarts club, Hermione has to get a finalized list of all interested people to submit to Dumbldore. Before combining and submitting the lists, she wants to check if any of the names repeat. Write a function that takes in the main list and the weekly one, checks to see if there are any repeated names, and returns a finalized list with no repeats. The returned memberList should have the same order as the interested list plus any recruits that were found not to be duplicates. Added recruits should be in the same order they appeared in the recruits list.

Note: The order of the interested list should take precedence when creating the final memberList.
Note: The interested list will not have duplicates.

```
>>> interested = ['Ron', 'Harry', 'Luna', 'Ginny']
>>> recruits = ['Ron', 'Cho', 'Luna']
```

```
>>> clubMembers(interested, recruits)
['Ron', 'Harry', 'Luna', 'Ginny', 'Cho']
```

```
>>> interested = ["George", "Angelina", "Parvati", "Lavender"]
>>> recruits = ["Ron", "Angelina", "Colin"]
>>> clubMembers(interested, recruits)
['George', 'Angelina', 'Parvati', 'Lavender', 'Ron', 'Colin']
```

Check Career

Function Name: checkCareer()
Parameters: students (`list`), career (`str`)
Returns: selectedStudents (`list`)
Description: McGonagall wants to see which of her fifth years want to do a certain job. Write a function that takes in a possible career path and a list of sublists. Each sublist will contain the student's name, and their desired career. The function should return a list of all the students that have the same desired career as the inputted career path. The list should contain the names in **alphabetical order**.

Hint: The .sort() method and sorted() function may be helpful in this question.

```
>>> students = [["Harry", "Auror"], ["Ron", "Auror"], ["Seamus", "Curse Breaker"]]
>>> checkCareer(students, 'Auror')
['Harry', 'Ron']
```

```
>>> students =  [["Ginny", "Quidditch Player"], ["Luna", "Professor"],
                 ["Padma", "Professor"]]
>>> checkCareer(students, 'Quidditch Player')
['Ginny']
```

High Grades

Function Name: highGrades()
Parameters: students (`list`), gpas (`list`)
Returns: honorsStudents (`list`)
Description: Dumbledore wants to throw a celebration for students at Hogwarts with a GPA of 3.5 or higher. He has a list of student names and a list of student GPAs. For each student in the student list, their corresponding GPA can be found at the same index in the GPA list. Write a function that takes in these two lists and returns a list of students who have a GPA of 3.5 or higher. If no students have a GPA of 3.5 or higher, return an empty list.

```
>>> students = ["Hermione", "Ron", "Ginny", "Fred", "Neville", "Draco"]
>>> gpas = [4.0, 2.8, 3.8, 2.3, 3.6, 3.4]
>>> highGrades(students, gpas)
["Hermione", "Ginny", "Neville"]
```

```
>>> students = ["Harry", "Dean", "Dudley", "Cho", "Luna"]
>>> gpas = [3.4, 3.5, 2.4, 3.2, 3.6]
>>> highGrades(students, gpas)
["Dean", "Luna"]
```

Quidditch Play

Function Name: quidditchPlay()
Parameters: playerOrder (`list`), partners (`list`)
Returns: isApproved (`bool`)
Description: Harry has come up with some plays to implement in the next quidditch game, involving a list of players in the order they will recieve the ball. However, there are some players that work best together and won't approve the play unless they pass the ball between them at least once. Write a function that takes in a list of names of players who are involved in the play and a list of two partner names and determine if one of them passes the ball to the other during the play (in either order). If they do, return True (`bool`), and if they don't, return False (`bool`).

Note: If one or both of the partners are not involved in the play, they will not approve of it, so in that case also return False (`bool`).

```
>>> playerOrder = ['George', 'Fred', 'Harry', 'George',  'Oliver', 'Alicia']
>>> partners = ['Fred', 'George']
>>> quidditchPlay(playerOrder, partners)
True
```

```
>>> playerOrder = ['George', 'Alicia', 'Charlie', 'Fred', 'Ron', 'Alicia',
                   'Harry', 'George']
>>> partners = ['Harry', 'Ron']
>>> quidditchPlay(playerOrder, partners)
False
```

Grading Rubric

Function	Points
messageDecoder()	20
clubMembers()	20
checkCareer()	20
highGrades()	20
quidditchPlay()	20
Total	**100**

Provided

The `HW04.py` skeleton file has been provided to you. This is the file you will edit and implement. All instructions for what the functions should do are in this skeleton and this document.

Submission Process

For this homework, we will be using Gradescope for submissions and automatic grading. When you submit your `HW04.py` file to the appropriate assignment on Gradescope, the autograder will run automatically. The grade you see on Gradescope will be the grade you get, unless your grading TA sees signs of you trying to defeat the system in your code. You can re-submit this assignment an unlimited number of times until the deadline; just click the “Resubmit” button at the lower right-hand corner of Gradescope. You do not need to submit your `HW04.py` on Canvas.